Efficient Exploration of Translation Variants in Large Multiparallel Corpora Using a Relational Database

Graën, Johannes ; Clematide, Simon ; Volk, Martin

Abstract: We present an approach for searching and exploring translation variants of multi-word units in large multiparallel corpora based on a relational database management system. Our web-based application Multilingwis, which allows for multilingual lookups of phrases and words in English, French, German, Italian and Spanish, is of interest to anybody who wants to quickly compare expressions across several languages, such as language learners without linguistic knowledge. In this paper, we focus on the technical aspects of how to represent and efficiently retrieve all occurrences that match the user's query in one of five languages simultaneously with their translations into the other four languages. In order to identify such translations in our corpus of 220 million tokens in total, we use statistical sentence and word alignment. By using materialized views, composite indexes, and pre-planned search functions, our relational database management system handles large result sets with only moderate requirements to the underlying hardware. As our systematic evaluation on 200 search terms per language shows, we can achieve retrieval times below 1 second in 75 % of the cases for multi-word expressions.

Efficient Exploration of Translation Variants in Large Multiparallel Corpora Using a Relational Database

Johannes Graën, Simon Clematide, Martin Volk
Institute of Computational Linguistics
University of Zurich
Zurich, Switzerland
{graen|siclemat|volk}@cl.uzh.ch

Abstract

We present an approach for searching and exploring translation variants of multi-word units in large multiparallel corpora based on a relational database management system. Our web-based application Multilingwis, which allows for multilingual lookups of phrases and words in English, French, German, Italian and Spanish, is of interest to anybody who wants to quickly compare expressions across several languages, such as language learners without linguistic knowledge.

In this paper, we focus on the technical aspects of how to represent and efficiently retrieve all occurrences that match the user's query in one of five languages simultaneously with their translations into the other four languages. In order to identify such translations in our corpus of 220 million tokens in total, we use statistical sentence and word alignment.

By using materialized views, composite indexes, and pre-planned search functions, our relational database management system handles large result sets with only moderate requirements to the underlying hardware. As our systematic evaluation on 200 search terms per language shows, we can achieve retrieval times below 1 second in 75 % of the cases for multi-word expressions.

Keywords: corpora, multiparallel, retrieval, database, evaluation

1. Introduction

In recent years, large parallel corpora have become popular not only for natural language processing but also for linguistic research and for language learners. Arguably, the most popular site is Linguee[1] which offers bilingual lexicon searches in combination with usage examples over word-aligned parallel corpora. These online systems have a number of shortcomings (Volk, Graën, and Callegaro, 2014). Most notably, they are restricted to bilingual searches. If a user is interested in a multilingual comparison, she must submit multiple queries.

On that account, we are developing a new corpus exploration tool to investigate translation variants in large multiparallel corpora. Our system *Multilingwis*[2] *(Multilingual Word Information System)* contains the texts of five languages from *Europarl*[3] with cross-language alignments down to the word level. Multilingwis allows the user to search for single words or multi-word expressions and returns the corresponding translation variants in the four other languages. Translation variants are all words and phrases that result from our statistical word alignment.

Corpus search systems for expert users require linguistic knowledge and information about the annotation layers, e.g. morphological symbols, part-of-speech tags, grammatical categories or how to infer the lemma given a word in a particular language. On the contrary, Multilingwis follows the principle of strict simplicity. The user types any word sequence as a query which is then interpreted by the system. First, the system determines the most likely language of the input words based on frequencies learned from the corpus. Then it strips the input sequence of all function words and triggers the query with the lemmas of all content words (adjectives, adverbs, nouns and verbs). Multilingwis retrieves all sentences with the search words in the given order where there are three or less function words in between any two search words. The challenge then lies in finding and highlighting the corresponding hits in the four target languages efficiently.

This paper first describes the preparation and linguistic annotation of our multiparallel corpus which is based on Europarl. We then describe in detail our technical solution for efficient retrieval based on advanced database techniques. Our evaluation shows that multiword retrieval for high-frequency input terms can be done efficiently even on large data sets.

2. Corpus Preparation

We extracted parallel text units[4] in English, French, German, Italian and Spanish from the *Corrected & Structured Europarl Corpus (CoStEP)*[5] (Graën, Batinic, and Volk, 2014) to each of which we subsequently applied the *TreeTagger* (Schmid, 1994) for tokenization, part-of-speech tagging and lemmatization. Tagging was done with the language models available from the TreeTagger's web page[6]. We adapted the TreeTagger's tokenizer (abbreviation lexicons, punctuation) and extended its tagging lexicon (especially the German one) with lemmas and part-of-speech tags for frequent words unknown to the language models.

[1] `http://www.linguee.com/`

[2] `http://pub.cl.uzh.ch/purl/multilingwis`

[3] `http://www.statmt.org/europarl/`

[4] Here, speaker turns from the sittings of the European Parliament.

[5] Altogether 146,652 speaker turns are available in all these five languages in CoStEP version 0.9.2, which bases on *Europarl* release v7 (Koehn, 2005). CoStEP is available at `http://pub.cl.uzh.ch/purl/costep/`.

[6] `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/#Linux`

Language-specific rules based on word forms, lemmas and part-of-speech tags allowed us to identify sentence segment boundaries, which separate parts of sentences by colon or semicolon[7]. After identifying sentence segments (about 1.7 million per language), we performed pairwise sentence alignment with *hunalign* (Varga et al., 2005) and based on that word alignment with *GIZA++* (Och and Ney, 2003; Gao and Vogel, 2008). Word alignment was performed on the lemmas[8] of content words for both directions on each language pair.

Having the corpus data processed as detailed above, we stored the data in a relational database as described in Graën and Clematide (2015). Our relational database management system (RDBMS) of choice is PostgreSQL[9] as it provides all the functionality that we rely on for our application.[10]

3. Efficient Retrieval from the Corpus Database

Since our retrieval method relies on lemmas both for source (query) and target (translation) languages, we built a *materialized view*[11] on lemmas including all relevant foreign keys so that this view comprises all relevant data and can be queried later on instead of the underlying tables. In case no lemma was given for a particular token[12], we include the word form instead as aforementioned. The view comprises one row (i.e. lemma tuple) for each original token which sums up to 220 millions over all languages and corresponds roughly to 44 million tokens per language.

We then built a *composite index* (see Winand, 2012, pp. 12–17) upon that view starting with the lemma itself and including all other columns in the order accessed by the query (lemma index) with the objective of not needing to fetch any actual data but the index when performing a corpus search. The index requires 7.3 GB of disk space which only adds 2.2 GB compared to an ordinary index over the lemma attribute of all 220 million rows.

In addition, we created a composite index on a symmetrized view of the word alignments (alignment index) that we had calculated. As symmetrization method we chose the union (Tiedemann, 2011, p. 76), thus favoring recall for our application. This index comprises 418 million single word alignments and requires 9 GB of disk space.

The search query first scans the lemma index in order to retrieve all matching token tuples within the same sentence segments for the search terms given. It then looks up the aligned tokens in all other languages by consulting the alignment index. Since we are not interested in the exact correspondence of lemmas from source to target languages but rather in the corresponding list of lemmas ordered by their appearance in the text, we can use the token tuple from the source language as a set when consulting the alignment index and hence the index gets scanned only once.

Subsequently, the query makes use of the lemma index again to retrieve lemmas for the tokens aligned which are concatenated to identify the particular *translation variant*.

For every reasonable count of search terms (up to nine words), we created a particular search function in the database in order for the database's query planner to already have a query plan (see Winand, 2012, pp. 172–179) prepared and, thus not needing to deal with it at runtime. Using several search functions, each one addressing a specific count of search terms, considerably outperforms a single function based on *Common Table Expressions (CTE)*[13] or recursion with a list of search terms as input.

Within these functions, we also count the frequencies of translation variants and rank the matching sentence segments of source and target languages by calculating a score that favors consistently short segments in all languages. These ones will be shown first in the example panel of the web application, depending on the user's selection of translation variants.

4. User-friendly Interface

We decided to build Multilingwis with a configuration-free web-based user interface. Upon entering one or more search terms, the system immediately gives feedback on the identified language and the accepted vs. ignored input words (i.e. content vs. function words). The query results appear quickly in the four other languages. They are sorted according to frequency and offer a number of options for the corpus exploration. See Clematide, Graën, and Volk (2016) for a description of the user interface.

In principle, every corpus sentence in the result set can be inspected. For many queries this is impractical because of the large number of hits. Therefore, Multilingwis allows the user to restrict the inspection to combinations of translation variants across languages. Given a German query, for example, the user may restrict the exploration to certain English and Spanish translation variants in combination. Particular variants for each language are hidden if they appear considerably less frequent than the most frequent variant, though the complete list can be checked by unfolding it.

Multilingwis helps investigating lexical variants in a single language by switching queries between languages. For instance, a German query results in a number of Spanish translation variants. By selecting one of those variants as a new query, one will get alternatives to the original German query. In this way, the languages may serve as mirrors for each other.

[7] More than 6 % of the segments in our corpus end with colon or semicolon.

[8] We used the word form instead if no lemma was provided; ambiguous lemmas are not disambiguated.

[9] `http://www.postgresql.org/`

[10] For a detailed feature comparison of major SQL databases see Winand (2012).

[11] Unlike regular views, materialized views are precalculated and thus provide faster access to the data queried in trade for disk space.

[12] These are mostly nouns that are unknown to the TreeTagger model.

[13] So called *Recursive Common Table Expressions* (they are not recursive themselves but their result sets can be understood as recursively defined) are a common way to iterate through list parameters. For our requirement, i.e. finding sentence segments given a list of search terms, a CTE would generate a first set of segments matching the first term and then incrementally build subsets of the respectively anterior set for every subsequent term in the list.

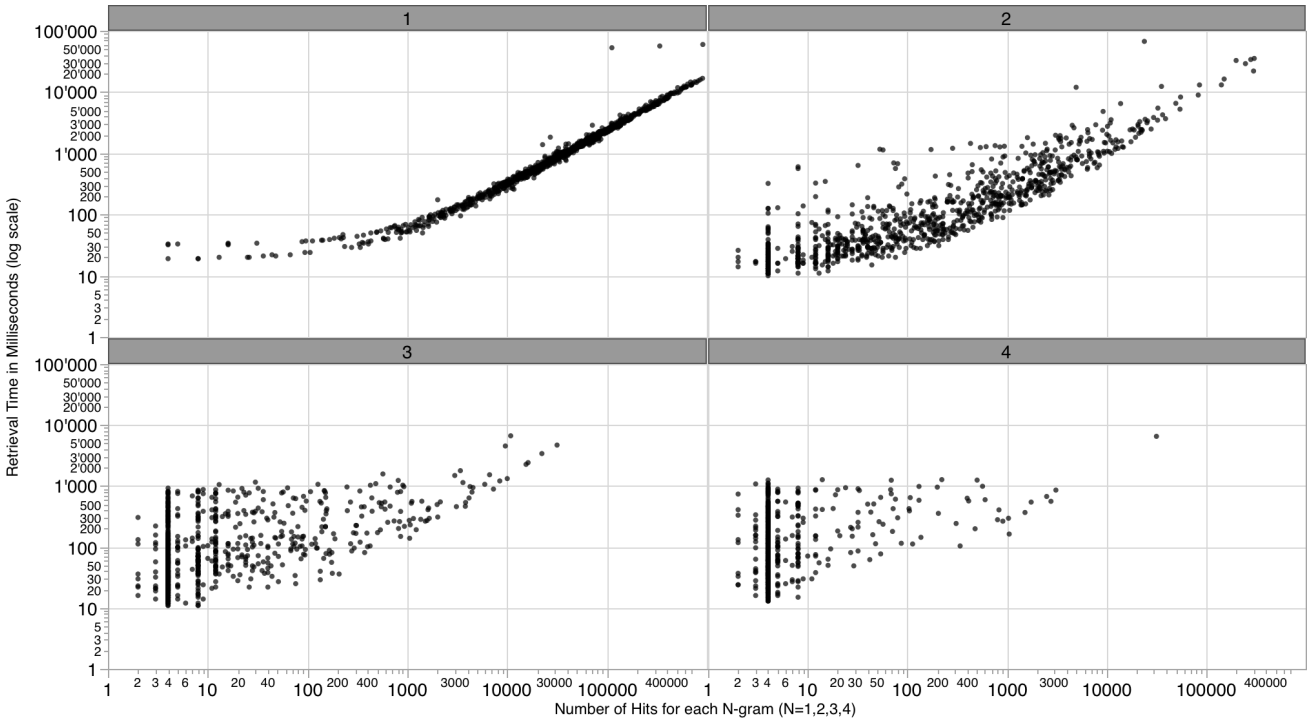

Figure 3: Correlation of the number of translation variants and retrieval time grouped per N-gram (N=1,2,3,4)

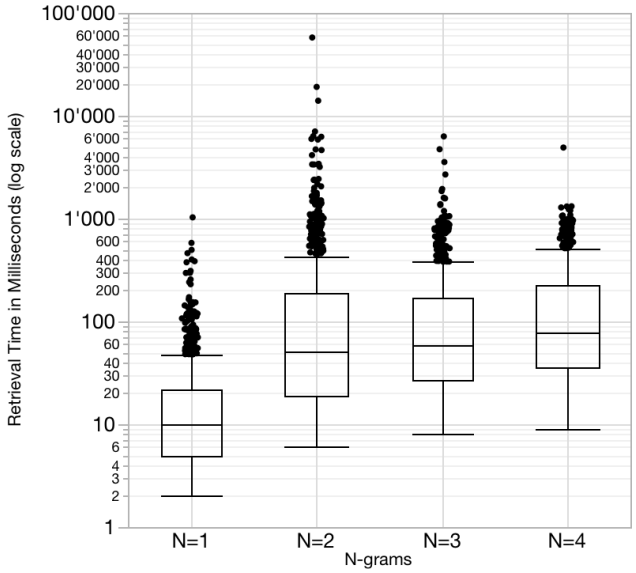

Figure 1: Boxplots of the retrieval time (ms) of all hits in the language of the query

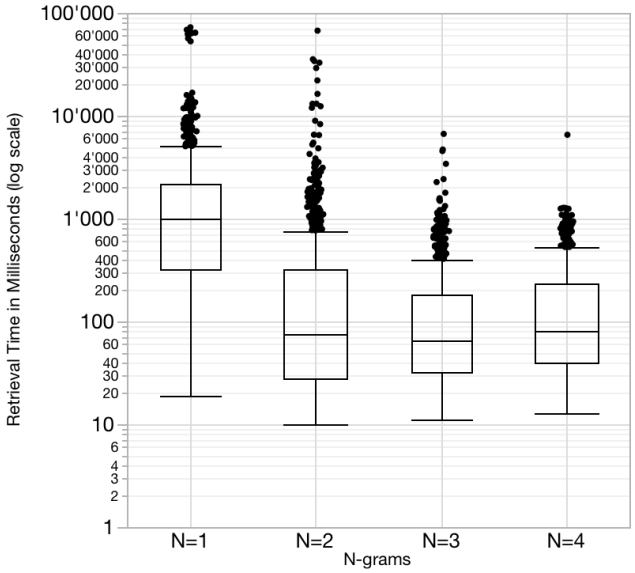

Figure 2: Boxplots of the retrieval time (ms) of all translation variants

5. Evaluation

In order to systematically evaluate the retrieval times of the database queries, we randomly sampled 200 different content lemmas from each language. These lemmas had to be followed by three additional content lemmas in the same sentence allowing for at most three intervening non-content words (proximity windows) between each content word. This experimental setup allows us to evaluate the retrieval times for n-grams of content words which share a common prefix, and, therefore, to assess whether the retrieval time for multi-word units decreases according to their frequency although each element of the multi-word unit might have a high frequency on its own.

All retrieval times were measured by a local PostgreSQL client performing the search on a dedicated Linux database host with PostgreSQL 9.5.0 (Intel Xeon E5-2650 2.6 GHz processors, SSDs for tablespace, 265 GB RAM). The numbers discussed report the time needed for retrieving the number of result rows (`SELECT count(*) FROM …`). For frequent words, the actual retrieval of the resulting rows

(`SELECT * FROM …`) can easily dominate the time needed for calculating that number.

Our first evaluation measures the time needed to find all hits in the language of the query. The boxplots in Fig. 1 show that the 75th percentile value is around 0.5 seconds or less for all languages. However, there are some outliers for combinations of frequent words where the retrieval time may take several seconds.

A further evaluation reports the time needed to retrieve all translation variants of all hits for a query, including the time to retrieve the hits in the language of the query. The boxplots in Fig. 2 show that the retrieval time for 4-gram multi-word units is dominated by the retrieval of the hits in the language of the query. For 4-grams, there are only a few hits in one language, and their translation variants can be found quickly. For 1-grams (single words), a substantial amount of computing time is needed in order to find all translation variants (up to 72 seconds for the highly frequent English verb "be"). However, the 75th percentile retrieval time for multi-word units is still below 1 second. As can be seen in Fig. 3, the correlation between the retrieval time and the number of translations decreases when the N of N-grams increases.

6. Conclusions

We implemented a corpus query system dedicated to the exploration of multi-word units in large multiparallel corpora based on a relational database management system (PostgreSQL).

In this paper, we discussed the technical implementation we chose in order to allow for an efficient retrieval of all translation variants for a given multi-word unit. Database indexes that are geared to the actual queries play a central role for fast retrieval.

Our evaluation shows that most multi-word queries (75 %) can be responded to within less than 1 second. Furthermore, the query response time decreases as the amount of words constituting the multi-word units increases.

7. Acknowledgment

This research was supported by the Swiss National Science Foundation under grant 105215_146781/1 through the project "SPARCLING – Large-scale Annotation and Alignment of Parallel Corpora for the Investigation of Linguistic Variation".

8. References

Clematide, S., J. Graën, and M. Volk (2016). "Multilingwis – A Multilingual Search Tool for Multi-Word Units in Multiparallel Corpora". In: *Computerised and Corpus-based Approaches to Phraseology: Monolingual and Multilingual Perspectives – Fraseología computacional y basada en corpus: perspectivas monolingües y multilingües*. Ed. by G. C. Pastor. Geneva: Tradulex, pp. 447–455.

Gao, Q. and S. Vogel (2008). "Parallel implementations of word alignment tool". In: *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*. Association for Computational Linguistics, pp. 49–57.

Graën, J., D. Batinic, and M. Volk (2014). "Cleaning the Europarl Corpus for Linguistic Applications". In: *Proceedings of the 12th KONVENS*. (Hildesheim), pp. 222–227.

Graën, J. and S. Clematide (2015). "Challenges in the Alignment, Management and Exploitation of Large and Richly Annotated Multi-Parallel Corpora". In: *3rd Workshop on the Challenges in the Management of Large Corpora*. (Lancaster). Ed. by P. Bański, H. Biber, E. Breiteneder, M. Kupietz, H. Lüngen, and A. Witt. Institut für Deutsche Sprache, pp. 15–20.

Koehn, P. (2005). "Europarl: A parallel corpus for statistical machine translation". In: *Machine Translation Summit*. (Phuket). Vol. 5. Asia-Pacific Association for Machine Translation (AAMT), pp. 79–86.

Och, F. J. and H. Ney (2003). "A Systematic Comparison of Various Statistical Alignment Models". In: *Computational linguistics* 29.1, pp. 19–51.

Schmid, H. (1994). "Probabilistic part-of-speech tagging using decision trees". In: *Proceedings of International Conference on New Methods in Natural Language Processing (NeMLaP)*. (Manchester). Vol. 12, pp. 44–49.

Tiedemann, J. (2011). "Bitext Alignment". In: *Synthesis Lectures on Human Language Technologies* 4.2, pp. 1–165.

Varga, D., P. Halácsy, A. Kornai, V. Nagy, L. Németh, and V. Trón (2005). "Parallel corpora for medium density languages". In: *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*. (Borovets), pp. 590–596.

Volk, M., J. Graën, and E. Callegaro (2014). "Innovations in Parallel Corpus Search Tools". In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*. (Reykjavik). European Language Resources Association (ELRA), pp. 3172–3178.

Winand, M. (2012). *SQL Performance Explained: Everything Developers Need to Know about SQL Performance*. Markus Winand.